

# Discriminative Training of Deep Fully-connected Continuous CRFs with Task-specific Loss

Fayao Liu, Guosheng Lin, Chunhua Shen\*

School of Computer Science, University of Adelaide, Australia

January 29, 2016

## Abstract

Recent works on deep conditional random fields (CRFs) have set new records on many vision tasks involving structured predictions. Here we propose a fully-connected deep continuous CRFs model for both discrete and continuous labelling problems. We exemplify the usefulness of the proposed model on multi-class semantic labelling (discrete) and the robust depth estimation (continuous) problems. In our framework, we model both the unary and the pairwise potential functions as deep convolutional neural networks (CNNs), which are jointly learned in an end-to-end fashion. The proposed method possesses the main advantage of continuously-valued CRFs, which is a closed-form solution for the Maximum a posteriori (MAP) inference.

To better adapt to different tasks, instead of using the commonly employed maximum likelihood CRFs parameter learning protocol, we propose task-specific loss functions for learning the CRFs parameters. It enables direct optimization of the quality of the MAP estimates during the course of learning. Specifically, we optimize the multi-class classification loss for the semantic labelling task and the Turkey's biweight loss for the robust depth estimation problem. Experimental results on the semantic labelling and robust depth estimation tasks demonstrate that the proposed method compare favorably against both baseline and state-of-the-art methods. In particular, we show that although the proposed deep CRFs model is continuously valued, with the equipment of task-specific loss, it achieves impressive results even on discrete labelling tasks.

## 1 Introduction

Recent works on combining conditional random fields (CRFs) and deep convolutional neural networks (CNNs) have set new records on many vision tasks involving structured predictions. Pixel-level labelling tasks generally refer to assigning a discrete or continuous label to each pixel in an image, with typical examples being semantic labelling, and depth estimation. They play a piv-

otal role in the computer vision community, and are fundamental to many high-level tasks, like recognition, scene understanding, 3D modeling. In the central of many pixel labelling tasks stands the feature engineering, which has a significant impact on the final results. However, designing effective and highly discriminative features remains a challenge. Traditional efforts have been focusing on designing hand-crafted features, *e.g.*, textron, HOG, SIFT, GIST, *etc.* Later on, unsupervised feature learning [1] has been introduced and showed promises. Recently, supervised feature learning, *i.e.*, deep CNNs have achieved tremendous success in various domains including image classification [2], pose estimation [3], semantic labelling [4–6], depth estimation [7,8], *etc.* Adapting CNNs for pixel labelling tasks have since attracted a lot of attention. However, several issues need to be resolved in this procedure, *e.g.*, down-sampled coarse predictions, non-sharp boundary transitions.

Besides the feature representation, another essential factor to be considered is the appearance and spatial consistency, which is typically captured by graphical models. Probabilistic graphical models, *e.g.*, CRFs [9], have long been shown as a fundamental tool in computer vision, and have been widely applied to structured prediction problems, like image denoising, semantic labelling, depth estimation, *etc.* Recently, continuous CRFs have shown benefits of efficient learning and inference, while being as powerful as discrete CRFs models [8,10–13]. In this paper, we present a general deep CRFs learning framework for both discrete and continuous labelling problems, which combines the benefits of CNNs and continuous CRFs.

While the maximum likelihood learning is dominant in learning CRFs parameters, it is argued being beneficial to directly take into account the quality of the MAP estimates during the course of learning. Samuel *et al.* [11] pointed out that if one's goal is to use MAP estimates and evaluate the results using other criteria, then optimization of the MRFs parameters using a probabilistic criterion, like maximum likelihood, may not necessarily lead to optimal results. Instead, a better strategy is to find the parameters such that the quality of the MAP estimates are directly optimized against a task-

\*Corresponding author (e-mail: chhshen@gmail.com).

specific empirical risk. For example, in image segmentation, the most direct criterion of a good segmentation mask is the global accuracy or the Jaccard index (intersection-over-union score). On the other hand, in many practical applications, it may not be necessary to model the full conditional distribution of labels, given observations. Instead, modeling the marginal conditional likelihood is more preferable. In [14], Domke argued that the marginal based loss generally yields better maximum posterior marginal (MPM) inference. Therefore an alternative to the maximum likelihood learning is to minimize the empirical risk loss. In [15], Kakade *et al.* propose to maximize the label-wise marginal likelihood rather than the joint label likelihood for sequence labelling problems. Here we analogously propose to maximize the label-wise labelling accuracy by optimizing the softmax classification loss for the semantic labelling task. For the depth estimation problem, depths captured by depth sensors are often noisy and contain missing values due to poor illumination and sensor limitation, *etc.* We tackle this problem by proposing a robust depth estimation model, which uses the robust Turkey’s biweight loss.

We summarize the main contributions of this work as follows.

- We present a general deep continuous CRFs learning model for both discrete and continuous pixel-level labelling problems. Both the unary and the pairwise potentials are modeled as CNNs networks, and jointly learned in an end-to-end fashion. There exists a closed-form solution for the MAP inference problem, largely reducing the complexity of the model. Our CRFs model is fully connected and thus may better capture long-range relations.
- Instead of the commonly used maximum likelihood criterion for CRFs parameter learning, we show that a task-specific loss function can be applied to directly optimize the MAP estimates, and show two different applications, namely, multi-class semantic labelling (discrete) and robust depth estimation (continuous).
- We experimentally demonstrate that the proposed framework shows advantages over the baseline methods and state-of-the-art methods on both semantic labelling and robust depth estimation tasks.

There seems to be a common understanding that for discrete labelling problems, continuous CRFs models may not be suitable. As a result, an increasingly sophisticated and computationally expensive discrete CRFs methods, in terms of both training and inference, have been developed in the literature. Here we argue that *a carefully-designed, yet much simpler, continuous CRFs model is capable of solving discrete labelling problems effectively and producing competitive results, at least in the case of deep CRFs*—CRFs with deep CNNs modeling the potentials.

## 1.1 Related work

In this section, we review some most recent progresses in CNNs based methods for semantic labelling and depth estimation tasks, which are closely related to our work.

**CNNs for semantic labelling** The most straightforward way of applying CNNs for semantic labelling is to design fully convolutional networks to directly output the prediction map, *e.g.*, [4,6]. These methods do not involve optimized structured loss, while only consider independent classification. A more appealing direction is to combine CNNs with graphical models, which explicitly incorporates structured constraints. A simple strategy to do so is to incorporate consistency constraints into a trained unary model as a post-processing step. These methods first train a CNNs model or generate CNNs features for constructing the unary potential, then add spatial pairwise constraints to optimize the CRFs loss. In the pioneering work of [16], Farabet *et al.* present a multi-scale CNNs framework for scene labelling, which uses CRFs as a post-processing step for local refinement. Most recently, Chen *et al.* [17] propose to first train a fully convolutional neural networks for pixel classification and then separately apply a dense CRFs to refine the semantic labelling results. Later on, more attempts are made towards joint learning of CNNs and graphical models. In the work of [5], Zheng *et al.* propose to implement the mean field inference in CRFs as recurrent neural networks (RNNs) to facilitate the end-to-end joint learning. Lin *et al.* [18] propose a piecewise training approach to learn the unary and pairwise CNNs potentials in CRFs for semantic labelling. They further propose in [19] to directly learn CNNs message estimators in the message passing inference rather than learning the potential functions. All of the above-mentioned joint learning approaches exploit discrete CRFs, which need to design approximation methods for both learning and inference. In contrast, we here explore continuous CRFs, which enjoys the benefits of exact maximum likelihood learning and closed-form solution for the MAP inference.

**CNNs for depth estimation** For the depth estimation task, Eigen *et al.* [7] propose to train multi-scale CNNs to directly output the predicted depth maps by optimizing the pixel-wise least square loss for depth regression. During the course of network training, there is no explicit structured constraints involved. Most recently, Liu *et al.* [8] propose a deep convolutional neural fields model for depth estimation from single images, which jointly learns continuous CRFs and deep CNNs in a single framework. They further propose a more efficient training approach based on fully convolutional networks and a superpixel pooling method in [13]. Our work here, which also jointly learn continuous CRFs and CNNs in a single framework, is mainly inspired by these two works [8,13]. However, our work differs from [8,13]

in the following important aspects. (a) Our method addresses both discrete and continuous labelling problems, while the works in [8,13] only deal with continuous labelling problem; Apart from continuous prediction, we extend continuous deep CRFs with closed-form MAP inference for discrete prediction. We show that by optimizing the task-specific empirical risk, it is possible to achieve competitive performance for discrete labelling tasks. (b) In contrast to the maximum likelihood optimization in [8,13], we show the flexibility and usefulness of employing a task-specific loss in the context of deep CRFs. (c) Our method here is based on fully-connected CRFs with both the unary and the pairwise potentials modeled as deep CNNs. In contrast, the pairwise term in [8,13] is a linear function (or single layer) built upon hand-crafted image features. In our model, the pairwise term is also a CNNs model, and all the features are learned end-to-end.

## 2 Fully-connected deep continuous CRFs

Fig. 1 shows an overview of our fully-connected deep continuous CRFs model, which consists of a unary network and a pairwise network. The unary network generates the unary prediction and is parametrized by  $\theta^U$ , while the pairwise network outputs a similarity matrix of superpixel pairs and is parametrized by  $\theta^V$ . The final prediction is performed by combining the unary and the pairwise outputs to conduct the MAP inference.

### 2.1 Discriminative learning of model parameters

Given  $N$  i.i.d. training examples  $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$ , CRFs typically learn the model parameters  $\theta$  by the maximum likelihood learning, *i.e.*, minimizing the negative conditional log-likelihood:

$$\min_{\theta} \frac{\lambda}{2} \|\theta\|_2^2 - \sum_{i=1}^N \log \Pr(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}; \theta), \quad (1)$$

where  $\frac{\lambda}{2} \|\theta\|_2^2$  is the regularization term, with  $\lambda$  being the weight decay parameter. The conditional log-likelihood is modelled as:

$$\Pr(\mathbf{y} | \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(-E(\mathbf{y}, \mathbf{x})), \quad (2)$$

where  $E$  is an energy function as described in Sec. 2.2;  $Z$  is the partition function defined as:  $Z(\mathbf{x}) = \int_{\mathbf{y}} \exp\{-E(\mathbf{y}, \mathbf{x})\} d\mathbf{y}$  for continuous labelling  $\mathbf{y}$ . During prediction, one performs the MAP inference:

$$\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y}} \Pr(\mathbf{y} | \mathbf{x}; \theta). \quad (3)$$

Instead of the traditionally optimized log-likelihood objective, we propose to directly optimize the quality of the estimated labelling (MAP solution). Similar to the discriminative max-margin framework proposed by Taskar *et al.* [20], our learning objective is:

$$\min_{\theta} \frac{\lambda}{2} \|\theta\|_2^2 + \sum_{i=1}^N \mathcal{L}(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}),$$

where  $\hat{\mathbf{y}}^{(i)} = \operatorname{argmax}_{\mathbf{y}} \Pr(\mathbf{y} | \mathbf{x}^{(i)}; \theta).$  (4)

Here  $\mathcal{L}(\cdot, \cdot)$  is a task-specific loss function, which measures the discrepancy of the predicted label with the ground-truth label;  $\mathbf{y}^{(i)}$ ,  $\hat{\mathbf{y}}^{(i)}$  are respectively the ground-truth and the predicted labels of the  $i$ -th training example. The loss function  $\mathcal{L}$  provides important information on how good a potential prediction  $\hat{\mathbf{y}}^{(i)}$  is with respect to the ground-truth label  $\mathbf{y}^{(i)}$ . This is generally not addressed in the traditional CRFs learning approaches that rely on the maximum likelihood scheme. To our knowledge, this has not been explored in the case of learning *deep* CRFs parameters. Note that the implicit differentiation technique can be applied to compute the gradient of  $\mathcal{L}(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)})$  with respect to the parameters  $\theta$ , which enables us to use stochastic gradient descent (SGD) to learn deep structured models.

### 2.2 Energy formulation

We now show how to generalize the deep continuous CRFs model in [13] to fully-connected multi-label discrete labelling problems, *e.g.*, multi-class image segmentation.

For the multi-class image segmentation task, given an image  $\mathbf{x}$ , we represent the label of each node (superpixel) as an  $m$  dimensional vector  $\mathbf{y}_p$  ( $\mathbf{y}_p \in \mathbb{R}^m$ ), which consists of the confidence scores of superpixel  $p$  being each of the  $m$  classes. Here  $m$  is the total number of semantic classes. Let  $\mathbf{y} = \mathbf{y}_1 \odot \mathbf{y}_2 \odot \dots \odot \mathbf{y}_n$  denotes the overall segmentation label of the image  $\mathbf{x}$ , with  $\odot$  stacking two vectors and  $n$  being the number of superpixels in  $\mathbf{x}$ . We formulate the energy function as a typical combination of unary potentials and pairwise potentials over the nodes (superpixels)  $\mathcal{N}$  and edges  $\mathcal{S}$  of the image  $\mathbf{x}$ :

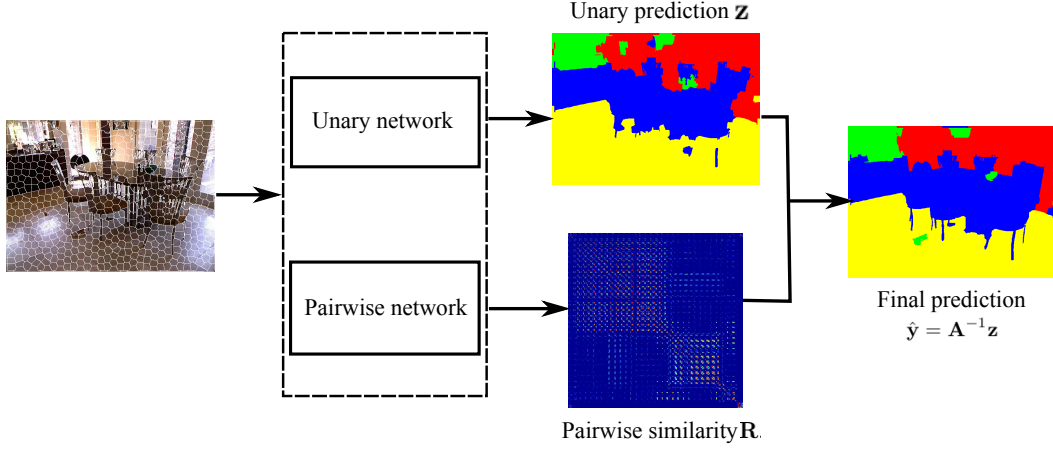
$$E(\mathbf{y}, \mathbf{x}) = \sum_{p \in \mathcal{N}} U(\mathbf{y}_p, \mathbf{x}) + \sum_{(p,q) \in \mathcal{S}} V(\mathbf{y}_p, \mathbf{y}_q, \mathbf{x}), \quad (5)$$

where  $U, V$  are the unary and pairwise potential function respectively.

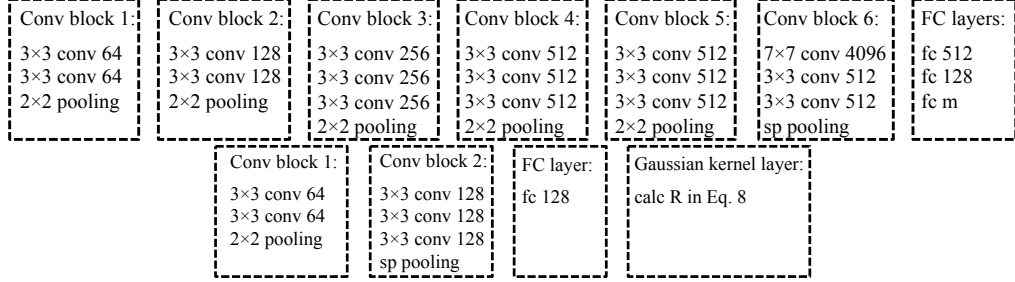
The unary potential is formulated as:

$$U(\mathbf{y}_p, \mathbf{x}; \theta^U) = \|\mathbf{y}_p - \mathbf{z}_p\|_2^2. \quad (6)$$

Here  $\mathbf{z}_p$  is the unary network (network architecture detailed in Fig. 2) output for the superpixel  $p$ , which is parametrized by  $\theta^U$ ;  $\|\cdot\|_2^2$  denotes the squared  $\ell_2$  norm of a vector.



**Figure 1:** The prediction framework of our method for the semantic labelling task. The image is first over-segmented into superpixels, and then as input to the unary and the pairwise networks. The unary network outputs the unary prediction of each superpixel, and the pairwise network outputs the pairwise similarities of superpixel pairs. The final prediction is obtained by combining the outputs of the unary and the pairwise networks to solve the MAP inference (Eq. (12)). Details of the unary and the pairwise networks are illustrated in Fig. 2.



**Figure 2:** The unary (upper plot) and the pairwise (lower plot) network architectures of our method. The unary network is composed of 6 convolution blocks and 3 fully-connected layers. The pairwise network is relatively shallow, which consists of 2 convolution blocks together with one fully-connected layer and a Gaussian kernel layer to calculate the pairwise similarity matrix  $\mathbf{R}$  in Eq. (8).  $m$  is the number of semantic classes. We use the superpixel pooling (denoted as sp pooling layer) method as proposed in [13] to obtain superpixel CNNs features from the convolution maps.

The pairwise potential is written as:

$$V(\mathbf{y}_p, \mathbf{y}_q; \mathbf{x}; \boldsymbol{\theta}^V) = \frac{1}{2} R_{pq} \|\mathbf{y}_p - \mathbf{y}_q\|_2^2. \quad (7)$$

Here  $R_{pq}$ , parameterized by the pairwise network parameters  $\boldsymbol{\theta}^V$ , is the output of the pairwise network (network architecture detailed in Fig. 2) from the superpixel pair  $(p, q)$ . Specifically,  $R_{pq}$  is computed from a Gaussian kernel layer:<sup>1</sup>

$$R_{pq}(\boldsymbol{\theta}^V) = \beta \exp(-\|\mathbf{s}_p - \mathbf{s}_q\|_2^2 - \gamma \|\mathbf{l}_p - \mathbf{l}_q\|_2^2), \quad (8)$$

where  $\mathbf{s}_p, \mathbf{s}_q$  are the convolution features (with dimension being 128 as illustrated in Fig. 2) for superpixel  $p, q$  respectively;  $\mathbf{l}_p, \mathbf{l}_q$  are the centroid coordinates of the superpixel  $p$  and  $q$  respectively (normalized to  $[0, 1]$ );  $\gamma$  is a hyperparameter that controls the scale of the Gaussian position kernel;  $\beta$  is a non-negative scaling factor, which is regarded as one of the pairwise network parameters and is jointly learned in our framework.

<sup>1</sup>Note that we are not limited to the Gaussian kernel here. We can use other similarity calculation forms such as Laplacian.

With Eqs. (6) and (7), the energy function in Eq. (5) can now be written as:

$$E(\mathbf{y}, \mathbf{x}) = \sum_{p \in \mathcal{N}} \|\mathbf{y}_p - \mathbf{z}_p\|_2^2 + \sum_{(p,q) \in \mathcal{S}} \frac{1}{2} R_{pq} \|\mathbf{y}_p - \mathbf{y}_q\|_2^2 \\ = \mathbf{y}^\top \mathbf{A} \mathbf{y} - 2\mathbf{z}^\top \mathbf{y} + \mathbf{z}^\top \mathbf{z}, \quad (9)$$

where

$$\mathbf{A} = \mathbf{A}_0 \otimes \mathbf{I}_m, \quad (10)$$

$$\text{with } \mathbf{A}_0 = \mathbf{I}_n + \mathbf{D} - \mathbf{R}. \quad (11)$$

Here  $\mathbf{z} = \mathbf{z}_1 \odot \mathbf{z}_2 \odot \dots \odot \mathbf{z}_n$ ;  $\mathbf{I}_m, \mathbf{I}_n$  are identity matrices of size  $m \times m, n \times n$  respectively;  $\otimes$  denotes the Kronecker product.  $\mathbf{R}$  is the  $n \times n$  matrix composed of  $R_{pq}$ ;  $\mathbf{D}$  is a diagonal matrix with  $D_{pp} = \sum_q R_{pq}$ .

## 2.3 Optimization

**MAP Inference** To predict the label of a new given image  $\mathbf{x}$ , we perform the maximum a posterior (MAP)



inference, which is:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \Pr(\mathbf{y}|\mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} E(\mathbf{y}, \mathbf{x}). \quad (12)$$

With the energy formulation in Eq. (9), we can obtain the closed-form solution for Eq. (12):

$$\hat{\mathbf{y}} = \mathbf{A}^{-1} \mathbf{z}, \quad (13)$$

where  $\mathbf{A}^{-1} = (\mathbf{A}_0 \otimes \mathbf{I}_m)^{-1} = \mathbf{A}_0^{-1} \otimes \mathbf{I}_m$ , with  $\mathbf{A}_0^{-1}$  can be obtained by solving a linear equation system. Note that for computational efficiency, one typically does not need to compute the matrix inverse explicitly.

*Discussion on MAP inference of fully connected CRFs* Note that even for fully connected models, the MAP inference here can still be calculated efficiently, due to the closed form of Eq. (13). In contrast in the case of discrete fully-connected CRFs models, efficient inference, even with various approximations, is only possible for a very special case to date. As indicated in [21], in order to apply the approximated fast inference of [21], the pairwise term must be in the form of the Gaussian kernel, and the input features must be in low dimension, typically less than 20. In our case, the input feature's dimension is 128, which is much higher than what can be possibly handled by [21].

### 2.3.1 Maximum likelihood learning

Since here we consider the continuous domain of  $\mathbf{y}$  (vector of confidence scores), the integral in the partition function in Eq. (2) can be analytically calculated under certain circumstances, as shown in [13]. According to [13], the negative conditional log-likelihood can be written as:

$$-\log \Pr(\mathbf{y}|\mathbf{x}) = \mathbf{y}^\top \mathbf{A} \mathbf{y} - 2\mathbf{z}^\top \mathbf{y} + \mathbf{z}^\top \mathbf{A}^{-1} \mathbf{z} - \frac{1}{2} \log(|\mathbf{A}|) + \frac{n}{2} \log(\pi). \quad (14)$$

The gradients of  $-\log \Pr(\mathbf{y}|\mathbf{x})$  with respect to  $\mathbf{A}$  and  $\mathbf{z}$  can be analytically calculated as shown in [13]. Then applying the chain rule (back propagation), the gradients with respect to  $\boldsymbol{\theta}^U$ ,  $\boldsymbol{\theta}^V$  and  $\beta$  can be calculated accordingly.

### 2.3.2 Task specific learning

**Multi-class semantic labelling** In the multi-class semantic labelling task, we obtain a prediction of  $m$  dimensional vector  $\hat{\mathbf{y}}_p$  for superpixel  $p$ , with each  $\hat{y}_{pj}$  ( $j$ -th element of  $\hat{\mathbf{y}}_p$ ,  $j = 1, 2, \dots, m$ ) indicating the confidence score of the superpixel  $p$  belonging to the  $j$ -th category. For the training ground truth  $\mathbf{y}_p$ , the confidence value for the ground truth class is 1, and 0 for other classes. During optimization, we prefer the confidence score of the ground-truth category being as large as possible compared to the confidence scores of incorrect

categories. In this circumstance, we minimize the softmax loss, typically employed in multi-class classification, i.e., maximize the label-wise classification accuracy, to directly pursue a desirable global accuracy:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = - \sum_{p \in \mathcal{N}} \sum_{j=1}^m y_{pj} \log \frac{\exp(-\hat{y}_{pj})}{\sum_{k=1}^m \exp(-\hat{y}_{pk})}. \quad (15)$$

Then after we get the prediction  $\hat{\mathbf{y}}_p$ , the predicted category label for the  $p$ -th superpixel is:

$$\hat{t} = \underset{j}{\operatorname{argmax}} \hat{y}_{pj}. \quad (16)$$

In the experiment section, we show that the use of this classification loss, instead of the traditional maximum likelihood loss, is critically important in achieving good performance for multi-class labelling tasks with our deep continuously-valued CRFs model.

**Robust depth regression** For the depth estimation problem, one of the main criteria for a good predictions is the root mean square (rms) error. Therefore, we can train our model by directly pursuing good MAP estimates in terms of RMS measure. To achieve this, the loss function for the depth estimation problem can be written as:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{p \in \mathcal{N}} \rho(r_p), \quad (17)$$

where  $r_p = y_p - \hat{y}_p$  is the depth residual of the superpixel  $p$  regarding to the ground truth;  $\rho(\cdot)$  is a penalty function, which generally satisfies  $\rho(0) = 0$  and  $\rho(r_p) > 0$  for  $r_p \neq 0$ .

A straightforward alternative to the log-likelihood loss is the least squares (LS) loss:

$$\rho(r_p) = r_p^2, \quad (18)$$

However, LS loss is generally not robust to outliers. Actually all convex regression loss functions are considered not robust. For example, for those extremely large residuals (typically outliers), the LS loss imposes too large penalty, which inevitably biases the model towards outliers.

To fully exploit the flexibility of the proposed task-specific CRF parameter learning, which allows us to choose arbitrary learning criterion, here we consider a non-convex, robust regression loss function, namely, the Turkey's biweight loss:

$$\rho(r_p) = \begin{cases} \frac{c^2}{6} [1 - (1 - \frac{r_p^2}{c^2})^3], & \text{if } |r_p| < c \\ \frac{c^2}{6}, & \text{otherwise.} \end{cases} \quad (19)$$

Here  $c$  is a pre-defined constant. Fig. 3 shows the Turkey's biweight loss and its first-order derivative.

	Ground	Structure	Furniture	Props	Pix. Acc.	Class Acc.
Gupta <i>et al.</i> [22]	-	-	-	-	78.0	-
Couprie <i>et al.</i> [23] (RGB+D)	87.3	86.1	45.3	35.5	64.5	63.5
Khan <i>et al.</i> [24]	87.1	88.2	54.7	32.6	69.2	65.6
Muller <i>et al.</i> [25]	<b>94.9</b>	78.9	71.1	42.7	72.3	71.9
Eigen <i>et al.</i> (RGB) [6]	-	-	-	-	75.3	-
Eigen <i>et al.</i> (RGB+D+N) [6]	93.9	87.9	79.7	55.1	80.6	79.1
Ours (unary)	84.3	84.8	77.3	54.6	77.1	75.8
Ours (full, RGB)	88.0	89.8	77.7	58.0	80.8	78.7
Ours (full, RGB+D)	92.9	<b>88.7</b>	<b>81.7</b>	<b>61.7</b>	<b>82.5</b>	<b>81.2</b>

**Table 1:** Segmentation results on the NYU v2 dataset (4-class). Note that our full model even outperforms the RGB+D+N model in [6] which uses additional ground-truth depth and surface normal information.

	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	blinds	desk	shelves	curtain	dresser	pillow	mirror	floor mat
Gupta <i>et al.</i> [26]	68.0	81.3	44.9	<b>65.0</b>	<b>47.9</b>	<b>47.9</b>	29.9	20.3	32.6	18.1	40.3	51.3	42.0	<b>11.3</b>	3.5	29.1	<b>34.8</b>	<b>34.4</b>	16.4	<b>28.0</b>
Eigen <i>et al.</i> [6]	<b>68.2</b>	<b>83.3</b>	45.0	51.9	46.0	40.4	<b>32.0</b>	17.6	33.0	31.5	45.0	<b>53.4</b>	43.9	<b>11.3</b>	<b>8.9</b>	<b>35.4</b>	26.3	31.0	<b>32.6</b>	27.4
Ours	67.7	75.9	<b>46.8</b>	52.2	43.8	46.3	29.0	<b>27.3</b>	<b>33.2</b>	<b>35.9</b>	<b>48.6</b>	41.9	<b>46.6</b>	9.0	7.8	32.9	28.9	24.6	18.1	23.0

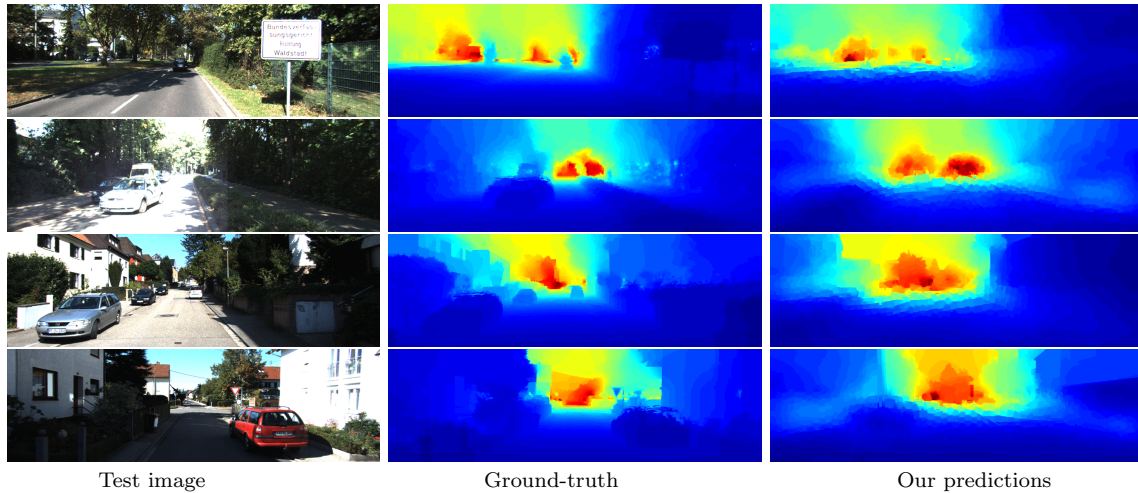
  

	clothes	ceiling	books	fridge	tv	paper	towel	s-curtain	box	w-board	person	n-stand	toilet	sink	lamp	bathtub	bag	other-struc	other-fur	other-prop
Gupta <i>et al.</i> [26]	4.7	60.5	6.4	14.5	31.0	14.3	16.3	4.2	2.1	14.2	0.2	<b>27.2</b>	<b>55.1</b>	37.5	<b>34.8</b>	<b>38.2</b>	0.2	7.1	6.1	23.1
Eigen <i>et al.</i> [6]	12.9	<b>75.2</b>	16.6	15.7	32.3	<b>20.3</b>	14.8	21.7	4.5	6.6	27.5	24.2	44.8	<b>39.4</b>	33.6	29.1	1.5	11.7	6.8	<b>29.4</b>
Ours	<b>15.6</b>	52.6	<b>19.3</b>	<b>22.3</b>	<b>40.7</b>	17.2	<b>22.5</b>	<b>24.1</b>	<b>5.6</b>	<b>46.9</b>	<b>47.6</b>	19.9	50.6	37.8	24.1	10.3	<b>4.2</b>	<b>13.4</b>	<b>10.4</b>	26.3

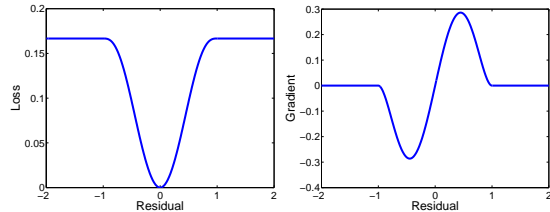
**Table 3:** The per-class pixel-wise Jaccard index for each of the 40 categories on the NYU v2 dataset. Our method performs the best on 18 out of the 40 classes. Note that Eigen *et al.* [6] use extra surface normal information, and their network takes multi-scale input images.

Noise level	Method	Error (lower is better)			Accuracy (higher is better)		
		rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
0%	Log loss	0.223	0.094	0.835	0.611	0.891	0.973
	Robust loss	<b>0.219</b>	<b>0.091</b>	<b>0.787</b>	<b>0.632</b>	<b>0.894</b>	<b>0.974</b>
10% noise	Log loss	0.263	0.115	1.023	0.518	0.815	0.945
	Robust loss	<b>0.222</b>	<b>0.092</b>	<b>0.790</b>	<b>0.628</b>	<b>0.896</b>	<b>0.975</b>
25% noise	Log loss	0.369	0.130	1.256	0.461	0.764	0.915
	Robust loss	<b>0.218</b>	<b>0.093</b>	<b>0.815</b>	<b>0.618</b>	<b>0.894</b>	<b>0.976</b>
10% outlier	Log loss	0.786	0.211	2.152	0.246	0.477	0.685
	Robust loss	<b>0.221</b>	<b>0.092</b>	<b>0.812</b>	<b>0.617</b>	<b>0.895</b>	<b>0.974</b>
25% outlier	Log loss	0.951	0.261	2.503	0.150	0.371	0.617
	Robust loss	<b>0.235</b>	<b>00.094</b>	<b>0.819</b>	<b>0.611</b>	<b>0.888</b>	<b>0.972</b>

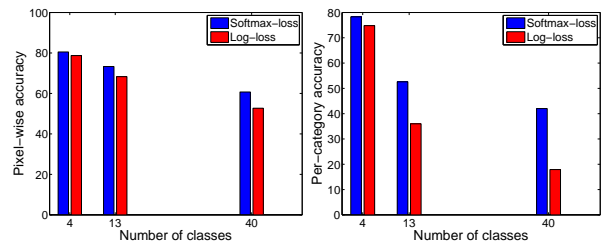
**Table 4:** Comparisons of depth estimation performance of different loss functions with respect to different noise/outlier ratios on the NYU v2 dataset.



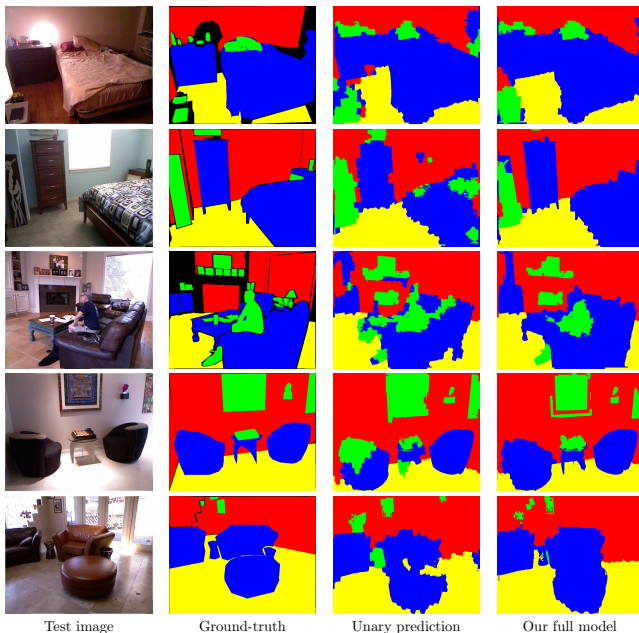
**Figure 5:** Examples of depth predictions on the KITTI dataset (Best viewed on screen). Depths are shown in log scale and in color (red is far, blue is close).



**Figure 3:** An illustration of the Turkey's biweight loss and its first order derivative.



**Figure 6:** Performance comparison of the conditional log-likelihood loss and the softmax loss for image segmentation on the NYU v2 dataset.



**Figure 4:** Semantic labelling (4-class) examples on the NYU v2 dataset. Compared to the unary predictions, our full model yields more accurate and smoother labelling.

	Pix. Acc.	Class Acc.	Avg. Jaccard	Freq. Jaccard
Gupta <i>et al.</i> [22]	59.1	28.4	27.4	45.6
Gupta <i>et al.</i> [26]	60.3	35.1	28.6	47.0
Khan <i>et al.</i> [24]	50.7	<b>43.9</b>	-	42.1
Long <i>et al.</i> [4] (RGB)	60.0	42.2	29.2	43.9
Long <i>et al.</i> [4] (RGB+D)	61.5	42.4	30.5	45.5
Eigen <i>et al.</i> [6] (RGB+D+N)	62.9	41.3	<b>30.8</b>	47.6
Ours (unary)	58.8	38.2	27.1	44.6
Ours (full, RGB)	62.6	42.2	30.5	<b>49.1</b>
Ours (full, RGB+D)	<b>63.1</b>	39.0	29.5	48.4

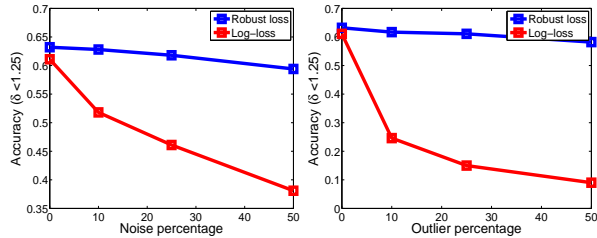
**Table 2:** State-of-the-art comparison on the NYU v2 dataset (40-class).

### 3 Experiments

To demonstrate the effectiveness of the proposed method, we perform experiments on the tasks of multi-class semantic labelling and robust depth estimation from single images. The first one is discrete-valued labelling and the second one is continuously-valued labelling.

**Implementation details** We implement our method based on the popular CNNs toolbox MatConvNet<sup>2</sup>. The network training is performed on a standard desktop with an NVIDIA GTX 780 GPU. We set the momentum as 0.9 and the weight decay parameter as  $\lambda = 0.0005$ . The first 5 convolution blocks and the first layer of the

<sup>2</sup><http://www.vlfeat.org/matconvnet/>



**Figure 7:** Prediction accuracies of the log-likelihood loss and the robust regression loss for depth estimation with respect to different percentages of noises/outliers on the NYU v2 dataset.

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena <i>et al.</i> [27]	0.280	-	8.734	0.601	0.820	0.926
Eigen <i>et al.</i> [7]	<b>0.190</b>	-	7.156	<b>0.692</b>	<b>0.899</b>	<b>0.967</b>
Liu <i>et al.</i> [13]	0.217	0.092	7.046	0.656	0.881	0.958
Ours	0.203	<b>0.086</b>	<b>6.427</b>	0.684	0.894	0.965

**Table 5:** Comparisons of depth estimation performance on the KITTI dataset. The results of [7] are obtained by using millions of training images while ours are obtained by training on 700 images.

6th convolution block of the unary network in Fig. 2 are initialized from the VGG-16 [28] model. All layers are trained using back-propagation. The learning rate for randomly initialized layers is set to  $10^{-5}$ ; for VGG-16 initialized layers it is set to a smaller value  $10^{-6}$ . The hyperparameter of the position Gaussian kernel is set to  $\gamma = 0.1$ . The parameter  $c$  in the Turkey’s biweight loss (Eq. (19)) is set to 1. We use SLIC [29] to generate  $\sim 700$  superpixels for each image.

### 3.1 Multi-class semantic labelling

The semantic labelling experiments are evaluated on the NYU v2 [30] and the MSRC-21 [31] datasets. The NYU v2 dataset consists of 795 images for training and 654 images for test (we use the standard training/test split provided with the dataset). For the semantic label ground-truth, we use the 4 and 40 classes described in [30], [22] respectively. The performance are evaluated using the commonly applied metrics as in [6,23,24,26], namely, pixel-wise accuracy (Pix. Acc.) and average per-class accuracy (Class Acc.). For the 40-class segmentation task, we also report the average Jaccard index (Avg. Jaccard) and the average pixel-frequency weighted Jaccard index (Freq. Jaccard) for each class. The MSRC-21 dataset is a popular multi-class segmentation benchmark with 591 images containing objects from 21 categories. We follow the standard split to divide the dataset into training/validation/test subsets.

**NYU v2** Since ground-truth depth maps are provided along with this dataset, we exploit depth cues as an additional setting (denoted as RGB+D). Specifically, we replicate the depth channel into 3 duplicates and regard them as images to input into our framework. The CNNs features from the RGB channels and the depth channels

are concatenated after the superpixel pooling layer. The compared results together with the per-class accuracy on the NYU v2 4-class segmentation task are reported in Table 1. As we can see, *our full model with incorporated depth cues outperforms state-of-the-art methods by a considerable margin*. It is worth mentioning that our method outperforms Eigen *et al.* [6], which used RGB images together with depths and surface normals. Compared with the unary-only model, our full model which jointly learns the unary and the pairwise networks shows consistent improvements in terms of all the metrics, demonstrating that the continuous CRFs model indeed improves discrete labelling. Fig. 8 shows some qualitative examples of our method together with the unary baseline. Compare to the unary predictions, our method yields more accurate and smoother labellings.

Table 2 summarizes the results on the 40-class segmentation task. Our method generally achieves better or comparable performance against state-of-the-art methods [4,6]. Specifically, with RGB only as the input, our method outperforms [4] using similar settings. Note that in [4], Long *et al.* achieved better results by using HHA encoding, which reveals that similar strategies can be applied here to further improve our results. In [6], Eigen *et al.* have used RGB images together with the ground-truth depths and surface normals as the input to their network training. Yet, our method achieves comparable performance. With depth channels incorporated, our method obtains the best global labelling accuracy. Our full model again shows improvements over the unary only model, demonstrating the usefulness of the joint learning scheme. The per-class pixel-wise Jaccard indexes of the 40 object categories are presented in Table 3, with our method achieves the best results on 18 out of the 40 classes.

**MSRC-21** Table 6 reports the segmentation results on the MSRC-21 dataset. As we can see, our results performs comparable to state-of-the-art methods [32], [33].

	Pix. Acc.	Class Acc.
Lucchi <i>et al.</i> [34]	83.7	78.9
Roy <i>et al.</i> [32]	91.5*	-
Liu <i>et al.</i> [33]	88.5	86.7
Liu <i>et al.</i> [33] (co-occur)	91.1*	<b>90.5*</b>
Ours (softmax loss, unary)	89.6	86.4
Ours (softmax loss, full)	<b>91.4</b>	89.0

**Table 6:** Segmentation results on the MSRC-21 dataset. Note that the results marked with \* are achieved by using additional mutex/co-occurrence information that models repelling relations.

**Ablation Study** To show the benefits of the multi-class classification loss over the tradition log-likelihood CRFs training, we perform ablation study on the performance comparisons of the two loss functions with respect to different numbers of classes. In addition to the 4-class and the 40-class segmentation tasks, we further perform experiments on a 13-class segmentation task by mapping

the 40 categories into 12 main classes and a ‘background’ class. Here the network structures are identical and the only difference is the loss function. Fig. 6 shows the comparative performance. We report both pixel-wise accuracy and per-category accuracy. As illustrated in the figure, *the multi-class soft-max loss considerably outperforms the log-likelihood loss*. The discrepancy becomes more significant when the number of classes increases. The previously-reported inferior performance of continuous CRFs on discrete labelling may be attributed to the use of the traditional maximum likelihood parameter learning.

### 3.2 Robust depth estimation

We perform robust depth estimation on two datasets: the indoor NYU v2 [30] and the outdoor KITTI [35] datasets. Several commonly used evaluation metrics, *i.e.*, relative error (rel), log10 error (log10), root mean square error (rms) and accuracy with different thresholds  $\delta$  (details are referred to [8,13]), are used here to report the results.

**NYU v2** The NYU v2 dataset [36] consists of 1449 RGB-D indoor image pairs, with 795 are used for training and 654 are for testing. We have used the standard training/test split provided along with the dataset.

In addition to performance comparisons on the original dataset, we also add additive white Gaussian noises or outliers to the ground-truth depths. For the noise case, we add additive white Gaussian noise with  $\sigma = 0.1$  to the  $[0, 1]$  normalized ground-truth depths. A certain percentage of pixels (*e.g.*, 10%, and 25% as shown in Table 4) are uniformly sampled to be added the noise. Likewise, for the outliers, we simply add a large value to the ground-truth of a certain percentage of uniformly sampled pixels. Table 4 reports the compared results of the log-likelihood loss (denoted as ‘log loss’) and the robust regression loss with respect to different noise/outlier levels. As we can observe, the robust regression loss is insensitive to both noises and outliers. In contrast, the performance of the log-likelihood loss degrades severely when the noise/outlier level increases. Fig. 7 illustrates the compared prediction accuracies of the two loss functions. The horizontal axis shows different noise (left plot) or outlier (right plot) percentages, and the vertical axis shows the depth prediction accuracy with  $\delta < 1.25$ .

**KITTI** The KITTI dataset [35] consists of outdoor videos taken from a driving vehicle equipped with a Li-DaR sensor. The ground-truth depths of the KITTI dataset are scattered at irregularly spaced points, which only consists of  $\sim 5\%$  pixels of each image, with all others unlabelled. We here treat those unlabelled regions as outliers and train the network using robust regression loss. The results are reported by using the same test set, *i.e.*, 697 images from 28 scenes, as provided by Eigen *et*

*al.* [7]. As for the training set, we use the same 700 images that Eigen *et al.* [7] have used to train the method of [27]. The compared results are reported in Table 5. It can be seen that our method outperforms the most recent work of [13] which have used the same training set, and very similar CNNs structure as ours. Compared to [7], which have used millions of training images, our results are very competitive with the lowest rms error achieved. Fig. 5 shows some prediction examples of our method. On both NYU v2 and KITTI datasets, our method using robust loss outperforms the log-likelihood loss, even with no noises deliberately added, because (a) original data may contain some noises more or less; (b) more importantly our method directly optimizes the task-specific empirical risk of interest.

We show some depth prediction examples on the NYU v2 dataset in Fig. 8. As we can see, our predictions well preserve depth discontinuities and align to local details.

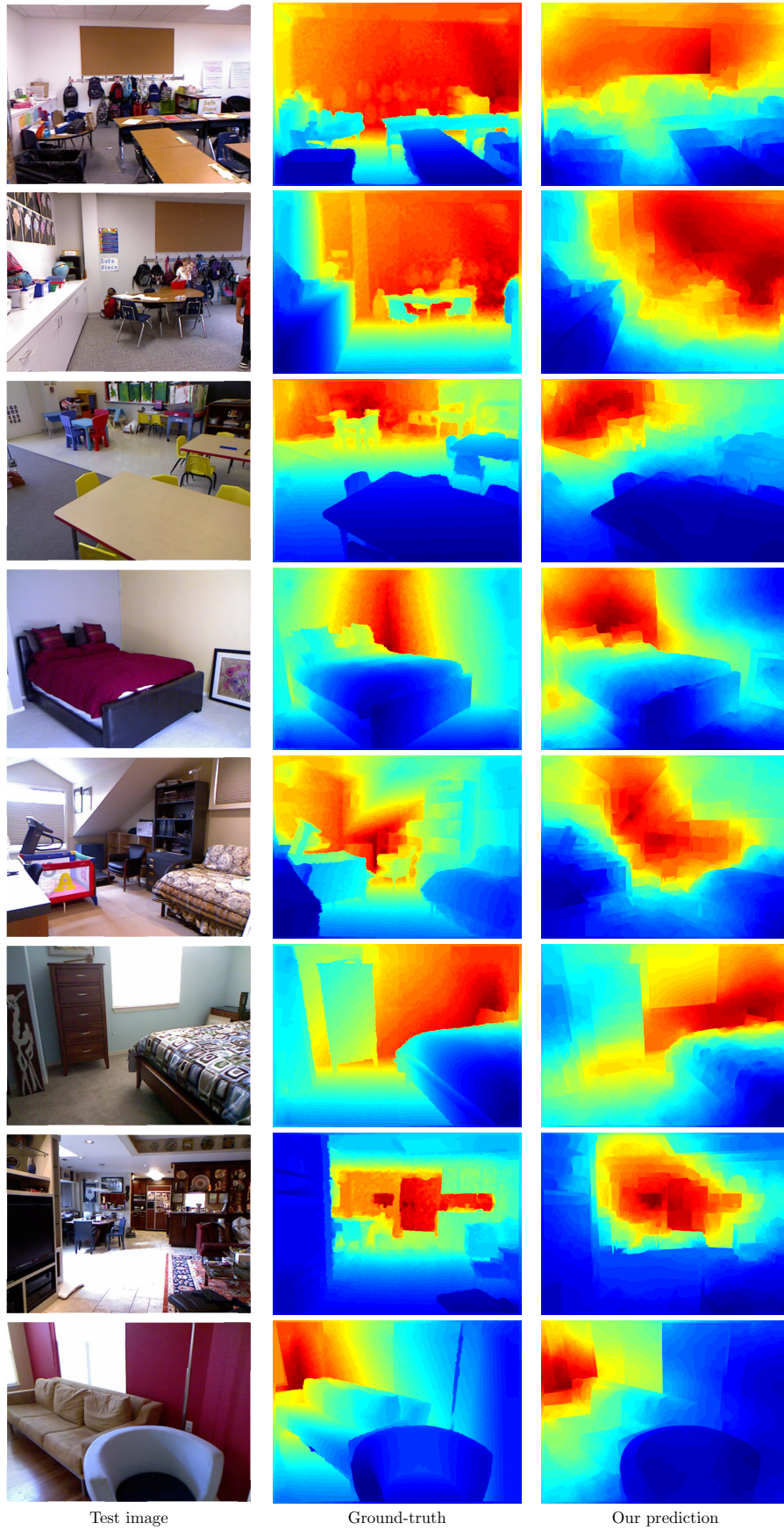
## 4 Conclusion

We have proposed a fully-connected deep continuous CRFs learning method for both discrete and continuous pixel-level labelling problems. The proposed method inherits the advantages of continuous CRFs, *i.e.*, exact maximum likelihood learning and closed-form solution for the MAP inference. Specifically, it models the unary and the pairwise potentials of the CRFs as two CNNs and learns the network parameters in an end-to-end fashion. For different tasks, we further propose to optimize a task-specific loss function, rather than the traditional log-likelihood loss in learning CRFs parameters, and demonstrate two applications, which are semantic labelling and robust depth estimation. Experimental results show that (a) the proposed method improves the unary-only model (standard regression without considering structured constraints), indicating the effectiveness of the pairwise term of our model; (b) our method achieves better or on par results compared with state-of-the-art, even on discrete labelling tasks, showing the promise of the proposed deep structured model. We thus advocate the use deep continuous CRFs in more structured prediction problems in vision, due to its power and in particular, significantly simpler and efficient inference, compared with most discrete deep CRFs such as [18,19].

## References

- [1] A. Coates, A. Y. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *Proc. Int. Conf. Artificial Intell. & Stat.*, 2011.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012.
- [3] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical





**Figure 8:** Depth prediction examples of our method on the NYU v2 dataset.

- model for human pose estimation,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.
- [4] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.
  - [5] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, “Conditional random fields as recurrent neural networks,” in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.
  - [6] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.
  - [7] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.
  - [8] F. Liu, C. Shen, and G. Lin, “Deep convolutional neural fields for depth estimation from a single image,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.
  - [9] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proc. Int. Conf. Mach. Learn.*, 2001.
  - [10] M. F. Tappen, K. G. G. Samuel, C. V. Dean, and D. M. Lyle, “The logistic random field - A convenient graphical model for learning parameters for mrf-based labeling,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2008.
  - [11] K. G. G. Samuel and M. F. Tappen, “Learning optimized MAP estimates in continuously-valued MRF models,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009, pp. 477–484.
  - [12] T. Baltrusaitis, P. Robinson, and L. Morency, “Continuous conditional neural fields for structured regression,” in *Proc. Eur. Conf. Comp. Vis.*, 2014.
  - [13] F. Liu, C. Shen, G. Lin, and I. D. Reid, “Learning depth from single monocular images using deep convolutional neural fields,” *CoRR*, 2015.
  - [14] J. Domke, “Learning graphical model parameters with approximate marginal inference,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.
  - [15] S. Kakade, Y. W. Teh, and S. T. Roweis, “An alternate objective function for markovian fields,” in *Proc. Int. Conf. Mach. Learn.*, 2002.
  - [16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.
  - [17] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” in *Proc. Int. Conf. Learn. Representations*, 2015.
  - [18] G. Lin, C. Shen, I. D. Reid, and A. van den Hengel, “Efficient piecewise training of deep structured models for semantic segmentation,” *CoRR*, vol. abs/1504.01013, 2015.
  - [19] G. Lin, C. Shen, I. D. Reid, and A. van den Hengel, “Deeply learning the messages in message passing inference,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015.
  - [20] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin, “Learning structured prediction models: a large margin approach,” in *Proc. Int. Conf. Mach. Learn.*, 2005.
  - [21] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected CRFs with Gaussian edge potentials,” in *Proc. Adv. Neural Info. Process. Syst.*, 2012.
  - [22] S. Gupta, P. Arbelaez, and J. Malik, “Perceptual organization and recognition of indoor scenes from RGB-D images,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2013.
  - [23] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, “Toward real-time indoor semantic segmentation using depth information,” *Int. J. Comp. Vis.*, 2014.
  - [24] S. Khan, M. Bennamoun, F. Sohel, R. Togneri, and I. Naseem, “Integrating geometrical context for semantic labeling of indoor scenes using rgbd images,” *Int. J. Comp. Vis.*, 2015.
  - [25] A. C. Müller and S. Behnke, “Learning depth-sensitive conditional random fields for semantic segmentation of RGB-D images,” in *Proc. IEEE Int. Conf. Robot. and Automation*, 2014.
  - [26] S. Gupta, R. B. Girshick, P. A. Arbeláez, and J. Malik, “Learning rich features from RGB-D images for object detection and segmentation,” in *Proc. Eur. Conf. Comp. Vis.*, 2014.
  - [27] A. Saxena, M. Sun, and A. Y. Ng, “Make3D: Learning 3d scene structure from a single still image,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.
  - [28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.
  - [29] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012.
  - [30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in *Proc. Eur. Conf. Comp. Vis.*, 2012.
  - [31] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, “*TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation,” in *Proc. Eur. Conf. Comp. Vis.*, 2006.
  - [32] A. Roy and S. Todorovic, “Scene labeling using beam search under mutex constraints,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014.
  - [33] F. Liu, G. Lin, and C. Shen, “CRF learning with CNN features for image segmentation,” *Pattern Recogn.*, 2015.
  - [34] A. Lucchi, Y. Li, and P. Fua, “Learning for structured prediction using approximate subgradient descent with working sets,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2013.
  - [35] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *Int. J. Robotic Res.*, 2013.
  - [36] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *Proc. Eur. Conf. Comp. Vis.*, 2012.